Feral Friends Final Report

Mark Bikakis, Hunter Brock, Tyler Woods

Capstone Project [Spring, 2020]

The University of West Florida

April 24, 2020

CIS 4595 & Capstone Systems Project

Dr. Bernd Owsnicki-Klewe

Abstract

Feral Friends is an android application developed by Tyler Woods, Hunter Brock, and Mark Bikakis. It aims to bring together community members through communication and collaboration to help colonies of feral cats. This application allows users to identify the location of feral colonies so others can view them and learn more about them. With the help of this app, the group hopes to improve the lives of these feral animals and to make it easier for people to identify and care for them.

Table of Contents

Table of Contents

1 Introduction

This report will explore the results of the Feral Friends capstone project. It will cover successes and failures, hurdles that were overcome, and potential work that could be done if the project was continued. It will also detail the extreme circumstances that impacted the project due to the COVID-19 pandemic.

2 Executive Summary

Feral Friends was a semester long project that required a variety of different skills to complete. Throughout the semester, the group worked to develop an android application to bring users together to help feral cats. Initial requirements were laid out to get a good idea of what is needed for the application. These requirements remained mostly intact in the final requirements but differed in the fine details. The initial timeline laid out a strict schedule that the group followed, however due to extreme circumstances, the final schedule was severely impacted. Additionally, initial expectations were set extremely high, and some of those expectations were met, while some were not. The project process helped the group to stay on track and continue to produce quality work throughout the semester. Finally, there is still plenty of work to be done to refine the application and to add additional features if it were to continue development after the semester is over.

3 Final Requirements VS Initial Requirements

The final requirements of the Feral Friends project are similar to the initial requirements but differ in fine details. The first final requirement is a system to allow users to login to the application. The group accomplished this by implementing Google's API and AWS Cognito allowing users to sign in using their Google account. This was chosen for multiple reasons. First, users are used to signing in using google accounts as it is widely used in other applications and websites. However, this can also be seen as somewhat of a negative as user accounts are tied to their central email. Google has very strong security, so this was not much of a worry for the group when making this decision. This requirement differs from the initial requirement of signing in because the group had originally planned on creating its own login system. The Google authentication route was chosen because it was more secure and worked better than designing a secure and functional sign in system for the application.

The second of the final requirements was posting a marker to the map. In the initial requirements, the group had laid out that users should be able to place a marker at the location of the colony and provide basic information like a name and description. Keeping these ideas in mind, additional functionality was added for this final requirement to make it better. More features were added when posting a marker like the ability to mark when the colony was fed, the number of cats in the colony, and an icon to show whether or not the colony had been TNR'ed. All of this information was stored for later use using AWS DynamoDB. By implementing AWS S3 into the platform, the group also gave users the ability to upload a picture of the colony for the markers. The pictures are stored in an S3 server for later use. These additions to the final requirement enhanced the functionality from the initial requirement allowing users to provide more information on the colonies.

The last of the final requirements was viewing markers on the map. In the initial requirement, the group laid out the barebones approach to displaying all the markers on the map for the users to view. Initially, the group followed the initial requirement but then later changed it in the most recent iteration of this final requirement. The group decided to allow users to edit markers that they view. This allows users to update a marker if they were to also help with that colony, incentivizing collaboration between users. Additional tools were implemented like AWS SNS, AWS Lambda and Firebase cloud messaging to enable the markers to be updated in real time for all users. When a marker is added or updated, Lambda handles the request. After the request is handled, AWS SNS uses Firebase cloud messaging to send a message to the application signaling it to update the markers from the database. This requirement was the most similar to its initial requirement counterpart as they are very straight forward.

Overall, the final requirements did end up similar to the initial requirements laid out in the project plan. However, the fine details of the final requirements differed quite a bit from the initial requirements. The changes made to these requirements throughout the development process added more to the application to make it better and easier to use.

4 Initial and Final Timelines

The group laid out a clear initial timeline. This timeline included the dates the sprints would begin and end. It also included important due dates like presentations and reports. This initial timeline greatly aided in planning out the tasks for the project. However, there were changes in the final timeline due to extreme circumstances.

4.1 Initial Timeline

1/31 - 2/14: First Sprint - Finish Planning/Documentation

2/11: Group Meeting

2/14 - 2/28: Second Sprint

3/03: Presentation 2

2/28 - 3/13: Third Sprint

3/13 - 3/27: Fourth Sprint

3/24 - 4/7: Individual Presentations

3/27 - 4/20: Preparation of Final Deliverables

4/21: Final Presentations

4.2 Final Timeline

The group followed the initial timeline very closely up until the end of the third sprint. Around this time the group encountered a major roadblock that severely impacted the expected timeline. In the early part of March, the University announced that it was moving all classes to remote instruction immediately following spring break due to the COVID-19 pandemic. This unprecedented situation caused the group major issues when it came to be working on the project for the majority of March. Group members scrambled to prepare for remote instruction for all of their classes and to deal with personal situations as a result of needing to follow strict guidelines to stay safe. On top of this, the group members dealt with increased stress as they tried to make up lost time in all of their classes as workloads increased due to the transition. During this time, group members attempted to continue to work on the project, but progress was severely impeded as a result of the situation. Finally, at the end of March/beginning of April, the group was able to begin getting back on track with the timeline of the project but were not fully able to get into the "groove" they had before spring break. The final timeline is shown below.

1/31 - 2/14: First Sprint - Finish Planning/Documentation

2/11: Group Meeting

2/14 - 2/28: Second Sprint

3/03: Presentation 2

2/28 - 3/13: Third Sprint

3/12 - 3/27: Complications due to COVID-19 pandemic

3/27 - 4/15: Fourth sprint and Individual Presentations

4/9 - 4/20: Continued Work on the Application and Preparation of Final Deliverables

4/21: Final Presentations

5 Project Results VS Expectations

Initially, the group had very high expectations for this project. The idea behind it had great potential and could supply a variety of very useful features to users. Unfortunately, the group learned quickly that they would not be able to meet the initial expectations they had set out. One of the high goals that was set early on was to publish the application on the google play store. This goal was infeasible due to multiple reasons. The first reason is due to the amount of work required. The polish and work needed to come up with an application that the group would feel comfortable publishing was not achievable during one semester while also balancing other classes and personal lives. On top of that, the resources required would exceed what was offered in the AWS free tier, which hosts the database the application is using. This goal was mainly there to inspire and motivate group members to reach for that high bar but knowing that they would probably never reach it in just one semester. Outside of that, the project resulted in an application that has the core functionality that the group envisioned during the project planning. The group was also expecting to use a more traditional web server and relational database approach but did not end up going in this direction. Instead they used AWS DynamoDB which gave them valuable experience in a rapidly growing platform. Given the COVID-19 situation, the group did the best they could to meet expectations despite the major setback. Overall, expectations were mostly met in the final result, but were not fully met due to losing valuable time because of the COVID-19 situation.

6 Project Process Review

The group followed a specific process that incentivized strong communication and consistent work. Following the Agile/SCRUM development cycle, the group would execute two weeklong sprints where they would tackle specific tasks. After these sprints, manual code review would be performed to look for major trust boundary issues in the application. They used Trello to track what needed to be done, what was being worked on, and what was already finished. Also using Trello, each task was assigned a number that represented its difficulty and importance. Doing this, the group was able to figure out which tasks and features were most important to the project so they could be prioritized. Discord was the main method of communication. Throughout the week group members would communicate using the text chat, but Friday evenings were the dedicated meeting time every week. During these meetings, members would discuss progress made, concerns that came up, and work on presentations and reports when needed. Discord greatly aided in being able to reach group members at any time, even during the period of complications due to COVID-19. This project process helped the project to go as smooth as it could, given the situation that happened this semester.

7 Work to Be Done

There is still plenty of work to be done with the Feral Friends project past this semester. If the group were to continue working on the project, additional testing would be done. Due to the schedule complication, there was not as much testing performed as the group would have liked to have done. Rigorous unit testing, user interface testing, security testing would need to be done before publishing the application to the play store. This would ensure that the app functions and behaves correctly, even under unusual or malicious circumstances. It would also help identify key points where the group could improve existing features.

The group would also like to add additional features and functionality to the application. There are three main features that they would like to add: group chats, AI to detect TNR, and expanding the ability to add pictures to the colony. First, improving the ability to add pictures would enhance the level of information a user could provide for the colony. Allowing the users to add multiple pictures or creating a timeline of pictures would be a great addition to the application. This would require additional backend work with both AWS S3 and AWS DynamoDB to accomplish this task. Next, group chats would greatly enhance the communication aspect of the application. This would allow groups of users to collaborate on helping colonies. It would enhance their ability to coordinate their care and ideas to help the animals. Lastly, the hardest feature to add would be implementing AI to detect TNR. TNR'ed cats have notched ears to signify that they have completed the process. Training AI to recognize these notched ears in pictures would require a large data set of these pictures and additional resources to accomplish the task. These three features would be what the group would work on implementing next if they were to continue development past this class.